

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A processor-implemented method of generating design rules for use in validating an electronic design ~~a rule-based operation~~, comprising the steps of:

establishing for each of a plurality of design rules for the electronic circuit design, a respective set of first, second, and third components;

wherein application of a design rule determines whether the electronic circuit design satisfies a requirement specified by the design rule;

wherein the first component of a design rule specifies triggering criteria that specify whether the design rule is applicable to the electronic design, the second component of the design rule specifies a condition, and the third component of the design rule specifies an action performed in response to the design rule being applicable based on the triggering criteria of the first component and the condition of the second component being satisfied;

~~dividing a design rule into at least one of three components, the three components comprising, a design rule triggering criteria, a condition, and an action, wherein the action is responsive to an evaluation of the condition;~~

wherein for each first, second, and third components of the plurality of design rules, the component is ~~expressing the design rule in at least one among of a datafile and source code~~, and the first, second, and third components are separate files;

for each first, second, and third component specified as a datafile, storing the datafile in a database;

for each first, second, and third component specified as source code, compiling the source code into object code;

and

binding each set of first, second, and third ~~the three~~ components together to form a rule object at runtime.

2. (Original) The method of claim 1, further comprising, entering at least one among a new design rule and a change to an existing design rule at runtime using a runtime

plug-in, wherein the step of entering is done without compiling.

3. (Original) The method of claim 2, wherein the step of entering at least one among the new design rule and the change to the existing design rule comprises the step of changing at least a portion of one component.

4. (Original) The method of claim 1, wherein the method further comprises forming a common object interface to a rule checker program to form an interface between the three components.

Claim 5. (Cancelled)

6. (Original) The method of claim 1, wherein the datafile includes an XML statement and the source code includes a C or C++ instruction.

Claims 7-10. (Cancelled)

11. (Currently Amended) The apparatus design-rule checker program of claim 20 ~~[[9]]~~, wherein the program further comprising ~~[[comprises]]~~ means for entering at least one among a new design rule and a change to an existing design rule at runtime using a runtime plug-in.

12. (Currently Amended) The apparatus design-rule checker program of claim 20 ~~[[9]]~~, wherein the program further comprising means ~~comprises a common object interface~~ for interfacing with a rule object formed from the first application criteria component, the second rule condition component, and the third action component at runtime.

13. (Currently Amended) The apparatus design-rule checker program of claim 20 ~~[[9]]~~, wherein the program further comprising ~~comprises a rule factory~~ means for building a rule repository for the design-rule checker program from at least one among

a datafile and program objects.

14. (Currently Amended) The apparatus design rule checker program of claim 13, further comprising means for searching for ~~wherein the rule factory searches~~ specified directories for user specified rule additions.

15. (Currently Amended) A machine readable storage, having stored thereon a computer program having a plurality of code sections executable by a machine for causing the machine to perform the steps of:

establishing for each of a plurality of design rules for the electronic circuit design, a respective set of first, second, and third components;

wherein application of a design rule determines whether the electronic circuit design satisfies a requirement specified by the design rule;

wherein the first component of a design rule specifies triggering criteria that specify whether the design rule is applicable to the electronic design, the second component of the design rule specifies a condition, and the third component of the design rule specifies an action performed in response to the design rule being applicable based on the triggering criteria of the first component and the condition of the second component being satisfied;

~~dividing any design rule into at least one of three components comprising an application criteria, a rule condition, and an action;~~

wherein for each first, second, and third components of the plurality of design rules, the component is ~~expressing a design rule in at least one among of~~ a datafile and an object source code, and the first, second, and third components are separate files; [[and]]

for each first, second, and third component specified as a datafile, storing the datafile in a database;

for each first, second, and third component specified as source code, compiling the source code into object code; and

binding each set of first, second, and third ~~the three~~ components together to form a rule object at runtime.

~~entering at least one among a new design rule and a change to an existing design rule at runtime using a runtime plug-in.~~

16. (Currently Amended) The machine readable storage of claim 15, wherein said step of entering binding is done without compiling.

17. (Currently Amended) The machine readable storage of claim 15, wherein the plurality of code sections further cause a machine to form a common object interface to a rule checker program to form an interface between the three components.

Claims 18-19. (Cancelled)

20. (New) A apparatus for generating design rules for use in validating an electronic design, comprising:

means for establishing for each of a plurality of design rules for the electronic circuit design, a respective set of first, second, and third components;

wherein application of a design rule determines whether the electronic circuit design satisfies a requirement specified by the design rule;

wherein the first component of a design rule specifies triggering criteria that specify whether the design rule is applicable to the electronic design, the second component of the design rule specifies a condition, and the third component of the design rule specifies an action performed in response to the design rule being applicable based on the triggering criteria of the first component and the condition of the second component being satisfied;

wherein for each first, second, and third components of the plurality of design rules, the component is one of a datafile and source code, and the first, second, and third components are separate files;

means, responsive to each first, second, and third component specified as a datafile, for storing the datafile in a database;

means, responsive to each first, second, and third component specified as source code, for compiling the source code into object code;

and

means for binding each set of first, second, and third components together to form a rule object at runtime.